

Designing with Computation-Realization Geometry

Certified Architecture Selection, Technology Matching, and Reference Physical Realization

Fitih M. Cinnor for semiAIfoundry Research

Research Lead and Accountable Author

semiAIfoundry Research, a research group of Semi AI Foundry LLC

Accountable correspondence: fcinnor@semiaifoundry.com

22 August 2026

Version 3.2.0

DOI: [10.5281/zenodo.22050638](https://doi.org/10.5281/zenodo.22050638) Version: 3.2.0

Executable companion: <https://github.com/semiaifoundry/computation-realization-geometry>

Collective scholarly attribution and research provenance. This work was produced by semiAIfoundry Research through a human-directed collaboration across the intelligence continuum—human, artificial, and hybrid. AI models and multi-agent systems made material contributions to mathematical exploration, formal derivation, computational implementation, verification, literature analysis, adversarial critique, and manuscript development. Fitih M. Cinnor led the research program and serves as the accountable human author, retaining responsibility for the scientific claims and corrections.

Abstract

This paper develops the design calculus of Computation–Realization Geometry. Technology enters after computation, legality, and attainability through feasible embeddings, service laws, and declared objectives. Finite algebraic registries induce semialgebraic technology phase cells with stable feasibility, active bottlenecks, class winners, ties, and margins. Inner constructive witnesses and scope-matched outer obligations form certified brackets and permit safe pruning. Argmin-hitting characterizes exact retention over a finite technology ledger, while price changes, embedding changes, and capability changes induce distinct reprice, re-embed, and reopen actions. The theory is connected to physical link count, serialization, energy, power, and shoreline laws; matrix multiplication, exact attention, state sharing, and ownership conversion; a working graph-to-certificate representation; and compact routed 2.5D/3D realizations. Computational-complexity limits of exact phase decomposition and minimum-cardinality ledger retention are stated explicitly. The resulting design discipline carries a selected realization together with its witnesses, bounds, margin, dependencies, and reopen conditions.

*Companion to **A Communication Theory of Computation Realization** (v3.2.0), DOI [10.5281/zenodo.22048090](https://doi.org/10.5281/zenodo.22048090). This manuscript expands the architecture-selection, technology-matching, certificate, and reference-physical-realization layers of the canonical work.*

Contents

1	Introduction	3
2	Relation to design methodology and optimization	3
3	Computation-realization geometry and safe commitment	4
I	Technology-conditioned realization and certified design	5
4	From attainable geometry to physical decisions	5
4.1	Pareto sufficiency and price-conditioned regimes	6
4.2	Certified inner and outer descriptions	7
4.3	Lossless retention for a technology domain	7

5	Architecture–technology scale and link matching	8
5.1	Scale-dependent communication strain	8
5.2	Aligned links and chiplet interfaces	9
6	Reference intermediate representation and compiler boundary	9
7	Analytic component certificates	10
7.1	Dense matrix multiplication across memory levels	10
7.2	A certified 2.5D technology map	11
7.3	Exact attention: recomputation versus storage	12
7.4	Cross-stage state and layout laws	13
II	End-to-end certificate	15
8	Certified nonseparable co-design of a long-context Transformer superblock	15
8.1	Workload and execution contract	15
8.2	Evidence-tiered technology contract	16
8.3	Finite architecture and mode registry	16
8.4	Conserved timing and energy	17
8.5	Strict registered decisions and phase walls	18
8.6	Context scaling and twelve-contract sensitivity	19
8.7	Evidence ledger and falsification conditions	20
III	Executable realization and end-to-end validation	21
9	From a compiler graph to physical consequence	21
9.1	Reference CRG intermediate representation	21
9.2	Head-to-head retention experiment	22
9.3	Carrying one semantic obligation into routed 2.5D and 3D models	23
9.4	Integrated computation-to-technology statement	24
10	Conclusion	24
	Collective attribution, accountability, and AI research provenance	29

1 Introduction

Architecture, dataflow, memory hierarchy, chiplet ownership, package service, and physical technology form one coupled decision. Modern DSE, compiler, and physical-design systems explore increasingly rich portions of that decision, but every exploration begins with a representation, a set of legal transformations, a candidate generator, an objective, and a technology model. The scientific problem is to make those precedences explicit and to determine which reduction of the candidate family preserves the decisions that remain.

This paper develops the design calculus of computation-realization geometry (CRG). A computation and resource contract define legal realizations and typed obligations before technology supplies embeddings, capacities, services, and prices. The geometry retains both constructive witnesses and scope-matched outer obligations. Technology then selects a realization from a stratified family indexed by hierarchy and contract.

The calculus has four layers. First, Pareto order and observation-matched geometry identify sufficient candidate boundaries. Second, finite semialgebraic registries produce complete phase cells carrying feasibility, active branches, bottlenecks, witness ties, and architecture ties. Third, inner/outer brackets give certified pruning, and argmin-hitting retention gives the exact finite-ledger analogue of the Safe-Commitment Law. Fourth, architecture-demand and technology-service scaling, aligned-link budgets, and physical embeddings connect the abstract signature to package and layout consequences.

The paper then composes these layers. Matrix multiplication, exact attention, state sharing, and layout conversion provide analytic component laws. A complete long-context Transformer registry produces technology-conditioned class reversals and phase walls. A PyTorch FX reference implementation builds contracts, legal realization classes, typed obligations, outer bounds, and decision certificates. A retention experiment tests which information survives post-commit changes. Routed compact 2.5D and 3D models carry the same semantic ownership boundary into package routing, timing, area, shoreline, power delivery, thermal, test, yield, and reliability coordinates.

The resulting object is a proof-carrying design discipline: one versioned realization geometry links semantic specification, compiler and architecture construction, technology selection, and physical evidence.

2 Relation to design methodology and optimization

Platform-based and contract-based design separate functional specification, architecture, and implementation through reusable assumptions and guarantees [1, 2, 3, 4]. Mead–Conway methodology and process design kits demonstrate the industrial power of moving coupled physical constraints into an executable upstream abstraction [5, 6]. CRG applies the same architectural principle before architecture selection: the resource contract defines legal transformations, the realization geometry records their joint obligations, and a decision certificate carries the conditions under which one alternative may be removed.

Compiler systems and intermediate representations generate legal program and schedule alternatives. The polyhedral model, Halide, TVM, MLIR, and equality saturation delay or organize selection among equivalent transformations [7, 8, 9, 10, 11, 12]. CRG adds terminal semantics, typed resource obligations, converse bounds, hierarchy structure, technology capability, and observation-sufficient retention. The reference IR in this paper is an executable instance of that interface.

Architecture DSE tools explore dataflows, mappings, memory hierarchies, parallel plans, and chiplet organizations [13, 14, 15, 16, 17, 18, 19]. MLDSE broadens multi-level hardware and mapping representations, and Mozart co-designs chiplet ecosystems and accelerators with physical validation [20, 21]. CRG complements these constructive engines by formalizing how the candidate space is constituted, which outer obligations apply to it, what information a reduction preserves, and when a capability change requires the legal family itself to be regenerated.

Vector and robust optimization supply Pareto order, upper images, scalarization, approximation, and parameter uncertainty [22, 23, 24, 25]. CRG’s semialgebraic phase cells and retention laws operate on a realization bundle rather than an anonymous objective image. This preserves schedule and embedding identity where a later physical objective distinguishes candidates with the same resource vector.

Physical EDA and architecture-level estimators evaluate selected witnesses through area, timing, energy, placement, and routing [26, 27, 28]. Open3DBench extends open backend evaluation to 3D partitioning, placement, routing, RC extraction, and thermal analysis [29]. In the CRG loop these systems are witness generators and refiners: their outputs enter the exact realization family with explicit evidence provenance and can change the decision geometry without changing the semantic definition of the computation.

Performance and accelerator co-design lineage. Roofline provides the canonical fixed-kernel compute–bandwidth balance [55]; Stream and Simba provide neighboring layer-fused and multi-chip-module accelerator

systems [56, 57]. CRG retains these as conditioned projections or realizations inside a prior legal-realization object.

3 Computation-realization geometry and safe commitment

For a computation F , resource contract ρ , and hierarchy $\mathcal{T}$, let $\mathcal{P}_F^\rho(\mathcal{T})$ be the legal realization family and $\sigma_{\mathcal{T}}$ its typed obligation map. The exact image $\mathcal{R}$ preserves joint attainability, $\Gamma = \text{cl up } \mathcal{R}$ preserves continuous monotone resource decisions, and $\mathcal{K} = \text{cl conv } \Gamma$ preserves additive nonnegative-price decisions. A retained family is safe for a declared decision class exactly when it preserves the corresponding object. Physical objectives that distinguish realization identity, schedule, timing, routing, or embedding must retain the richer realization fiber.

This ordering is essential. A technology price change can select a different point in an unchanged legal family; a technology capability that enables a new primitive, storage mode, hierarchy, or ownership structure changes the family itself.

Part I

Technology-conditioned realization and certified design

4 From attainable geometry to physical decisions

The preceding parts construct the legal realization family and its decision-sufficient geometries before physical technology is applied. This part completes the computation-to-technology chain. A technology supplies capabilities, admissible embeddings, service laws, and costs; the design problem is to determine which legal realization and embedding minimize the declared physical objective while preserving every alternative required by the future decision class.

CRG and existing design tools occupy complementary mathematical roles. Timeloop/Accelergergy, MAESTRO, ZigZag, CoSA, Gemini, and physical implementation flows generate, evaluate, and refine constructive witnesses [13, 26, 14, 15, 16, 19]. CRG supplies the computation-conditioned realization bundle, typed joint obligations, scope-matched converses, and exact retention conditions under which those witness sets may be reduced. The combined flow is closed-loop: search and physical tools return realizations to the geometry, and the geometry determines which further searches and physical measurements can change the decision.

Fix a terminalized computation F , a resource contract ρ , and an admissible hierarchy $\mathcal{T}$. Let $\mathfrak{R}_F^\rho(\mathcal{T})$ be the legal-realization space and $\sigma_{\mathcal{T}}(P) = z = (u, \mu)$ its typed signature, where u records local obligations and μ records interface state, traffic, causal, and activity coordinates. Different realizations can share one signature, so a detailed physical objective need not factor through z .

Definition 4.1 (Technology-conditioned physical objective). *For technology τ , let $\text{Emb}_\tau(P, \mathcal{T})$ be the legal embeddings of realization P . Define*

$$\text{OPT}_\tau^\rho(F) = \inf_{\substack{\mathcal{T} \in \text{Dec}_\rho(F), P \in \mathfrak{R}_F^\rho(\mathcal{T}), \\ \phi \in \text{Emb}_\tau(P, \mathcal{T})}} \Phi_\tau(P, \phi).$$

When the objective observes only the typed signature and embedding, write $\Phi_\tau(P, \phi) = \widehat{\Phi}_{\tau, \mathcal{T}}(\sigma_{\mathcal{T}}(P), \phi)$. Timing, routing, clocking, switching, thermal, and reliability objectives may distinguish realizations in the same signature fiber.

Definition 4.2 (Price update and capability update). *Fix a computation F , a resource contract ρ , and a technology $\tau = (\text{Emb}_\tau, \Phi_\tau)$. A price update replaces Φ_τ by another objective $\Phi_{\tau'}$ while preserving the legal realization family and the embedding relation:*

$$\mathcal{P}_F^\rho(\mathcal{T}) \text{ unchanged,} \quad \text{Emb}_{\tau'}(P, \mathcal{T}) = \text{Emb}_\tau(P, \mathcal{T}).$$

A capability update changes an allowed primitive, hierarchy, storage mode, schedule, carrier, or embedding relation and therefore may replace ρ by ρ' or Emb_τ by a different admissibility relation.

Proposition 4.3 (Repricing and branch reopening). *Under a price update, the pre-technology CRG objects $\mathcal{P}, \mathcal{R}, \Gamma$, and $\mathcal{K}$ are unchanged; the updated decision is obtained by re-evaluating the new objective on the retained object sufficient for its observation class. Under a capability update, the legal realization or embedding family may change. A retention certificate transfers to the updated problem exactly when the updated legal family contributes no additional decision-relevant point and the new objective remains inside the observation class preserved by the retained family.*

Proof. A price update changes neither legality nor attainable signatures, so the abstraction ladder is identical and only the response functional changes. A capability update can add or remove realizations or embeddings, hence can change the exact image and every downstream quotient. If the updated family adds no point that improves a decision in the preserved class, the relevant quotient remains equal and the Safe-Commitment Law transfers. Conversely, if the updated quotient differs, its reconstruction theorem supplies a future monotone objective or nonnegative price that distinguishes the old and new families; the affected branch must be reopened. $\square$

Across hierarchies, signature dimensions and legality rules differ. The global design space is therefore the stratified disjoint union

$$\mathfrak{D}_F^\rho = \bigsqcup_{\mathcal{T} \in \text{Dec}_\rho(F)} \{\mathcal{T}\} \times \mathfrak{R}_F^\rho(\mathcal{T}),$$

not one global convex body. Convex support arguments apply within a fixed hierarchy and observation class; cross-hierarchy selection retains the stratum label and its physical embedding constraints.

4.1 Pareto sufficiency and price-conditioned regimes

Proposition 4.4 (Signature dominance). *Fix a hierarchy and an objective that factors through the signature and is coordinatewise nondecreasing. If $z_1 \leq z_2$, then z_2 cannot have lower cost than z_1 . Hence only the Pareto-minimal signature boundary can be preferred by that observation class.*

Proof. Monotonicity gives $\widehat{\Phi}_{\tau, \mathcal{T}}(z_1, \phi) \leq \widehat{\Phi}_{\tau, \mathcal{T}}(z_2, \phi)$ whenever both signatures are evaluated under the same admissible embedding conditions. The claim does not compare realization fibers when the objective observes more than z . $\square$

For a fixed hierarchy and a nonnegative additive price vector w , a point z of the convex upper body $\mathcal{K}_F^o(\mathcal{T})$ is optimal exactly when w lies in its lower normal cone

$$N_{\mathcal{K}}^-(z) = \{w \geq 0 : \langle w, y - z \rangle \geq 0 \text{ for all } y \in \mathcal{K}\}.$$

Changing prices changes the exposed face without changing the computation. Integer allocations, feasibility walls, and maxima of compute, memory, fabric, and bridge time require a more general finite phase object.

Definition 4.5 (Finite semialgebraic architecture registry). *Let $\Theta \subseteq \mathbb{R}^p$ be a semialgebraic technology domain. A finite registry consists of architecture classes $a \in \mathcal{A}$ and finite witness sets W_a . A witness $\omega \in W_a$ records every discrete realization, allocation, layout, and schedule choice. Its feasibility is described by finitely many polynomial or rational inequalities whose denominators have fixed sign on Θ ; its cost is assembled from rational functions by finitely many min and max operations. Infeasible witnesses have cost $+\infty$. Put*

$$C_a(\theta) = \min_{\omega \in W_a} C_{a\omega}(\theta), \quad A(\theta) = \left\{ a : C_a(\theta) = \min_b C_b(\theta) < +\infty \right\}.$$

If every registered class is infeasible, $A(\theta) = \emptyset$.

Theorem 4.6 (Finite technology-phase stratification). *Every registry satisfying Theorem 4.5 admits a finite partition of Θ into connected, relatively open semialgebraic sign cells, with relative openness taken in each cell's supporting stratum. On every cell the following complete set-valued labels are constant:*

- (i) feasible classes and witnesses;
- (ii) every active rational branch and bottleneck set, including ties;
- (iii) the optimal witness set within every class; and
- (iv) the globally winning architecture set $A(\theta)$.

Every label change lies on a feasibility wall, branch or bottleneck exchange, witness exchange, architecture tie, or boundary of Θ .

Proof. Clear denominators on regions where their signs are fixed. Collect the finite family of feasibility polynomials, branch-comparison polynomials, within-class and cross-class cost differences, and defining polynomials of Θ . A sign-invariant semialgebraic cell decomposition exists for this finite family [42]; refine each stratum into connected cells. Every comparison then has one constant sign in $-, 0, +$. Retaining the zero sign preserves complete tied branch, bottleneck, witness, and architecture sets instead of inserting a hidden tie-break. The listed labels are consequently constant, and they can change only when a defining sign or domain stratum changes. $\square$

Corollary 4.7 (Certified full-label phase radius). *Let $\mathcal{P}$ contain every cleared feasibility polynomial, branch difference, within-class witness difference, and cross-class difference used in Theorem 4.6. Suppose $\theta_0 \in \Theta$, every $p \in \mathcal{P}$ is nonzero at θ_0 , and p is L_p -Lipschitz on $\overline{B}(\theta_0, r_0) \cap \Theta$. Then the complete phase label is unchanged throughout $B(\theta_0, r) \cap \Theta$ whenever*

$$r < \min \left\{ r_0, \min_{p \in \mathcal{P}: L_p > 0} \frac{|p(\theta_0)|}{L_p} \right\},$$

where the inner minimum is $+\infty$ if every $L_p = 0$.

Proof. For $|\theta - \theta_0|_2 < |p(\theta_0)|/L_p$, the Lipschitz inequality prevents p from reaching zero, so every defining sign is preserved. Intersecting with Θ also covers a point on the admissible-domain boundary. The radius controls losing-class witness exchanges and presently infeasible witnesses as well as the global winner. $\square$

A one-axis distance to a tie wall is therefore only a directional diagnostic; it is not the full-label radius unless every defining sign and the geometry of Θ have been included.

Computational complexity of exact phase and retention objects. The finite semialgebraic partition is an existence result, not a polynomial-time guarantee: general real quantifier elimination and cylindrical algebraic decomposition can be doubly exponential in the number of variables [61]. Likewise, finding a minimum-cardinality retained family that intersects every declared argmin set is a set-cover problem and is NP-hard in general [60]. Practical systems therefore use restricted parameterizations, incremental ledgers, certified bounds, and approximation algorithms while preserving certificate scope.

4.2 Certified inner and outer descriptions

Definition 4.8 (Certified outer relaxation). *For each hierarchy let*

$$\text{Adm}_\tau(\mathcal{T}) \subseteq \{(P, \phi) : P \in \mathfrak{R}_F^p(\mathcal{T}), \phi \in \text{Emb}_\tau(P, \mathcal{T})\}$$

be the exact admissible pairs. A lower-bounding relaxation consists of a superset

$$\text{Adm}_\tau^{\text{out}}(\mathcal{T}) \supseteq \{(\sigma_\tau(P), \phi) : (P, \phi) \in \text{Adm}_\tau(\mathcal{T})\}$$

and an optimistic objective satisfying

$$\Phi_{\tau, \mathcal{T}}^{\text{out}}(\sigma_\tau(P), \phi) \leq \Phi_\tau(P, \phi)$$

for every exact admissible pair.

Theorem 4.9 (Certified sandwich and safe pruning). *Let $\mathcal{J}_\in$ be a finite set of actual legal realization–embedding pairs. Then*

$$\boxed{\text{LB}_\tau \leq \text{OPT}_\tau^p(F) \leq \text{UB}_\tau,}$$

where

$$\text{LB}_\tau = \inf_{\mathcal{T}} \inf_{(z, \phi) \in \text{Adm}_\tau^{\text{out}}(\mathcal{T})} \Phi_{\tau, \mathcal{T}}^{\text{out}}(z, \phi), \quad \text{UB}_\tau = \min_{(P, \phi) \in \mathcal{J}_{\text{in}}} \Phi_\tau(P, \phi).$$

A candidate class may be discarded whenever its scope-matched certified lower bound is strictly larger than the constructive incumbent.

Proof. Every exact admissible pair maps into the relaxed set with no larger relaxed cost, so the relaxed infimum is a lower bound. Every member of $\mathcal{J}_\in$ is an exact feasible design, so its best evaluated cost is an upper bound. A class whose lower bound exceeds that incumbent cannot contain an in-scope optimizer. $\square$

The scope check is part of the proof obligation. Computation, candidate class, objective and units, hierarchy, resource contract, technology contract, assumptions, and validity domain must match the decision being made. Any dependency change invalidates the affected certificate and reopens the corresponding branch. The retained object is the bracket and its dependency ledger, not merely the lowest evaluated point.

4.3 Lossless retention for a technology domain

Theorem 4.10 (Argmin-hitting retention). *Let $\mathcal{P}$ be a finite registered realization set and let Θ be any technology domain on which every technology has a finite attained optimum. Define*

$$\mathcal{M}_\theta = \underset{P \in \mathcal{P}}{\text{argmin}} C_\theta(P).$$

A retained family $\mathcal{A} \subseteq \mathcal{P}$ preserves the optimum at every technology in Θ if and only if

$$\boxed{\mathcal{A} \cap \mathcal{M}_\theta \neq \emptyset \text{ for every } \theta \in \Theta.}$$

A singleton is lossless exactly when $\bigcap_{\theta \in \Theta} \mathcal{M}_\theta \neq \emptyset$.

Proof. At one technology, the retained and full minima agree exactly when a retained realization attains the full minimum. Quantifying this equivalence over Θ proves both statements. $\square$

This is the exact statistic for a named technology domain. It is weaker than universal preservation for all continuous monotone objectives, which requires preserving the complete upper obligation region, and weaker than preservation for every additive nonnegative price, which requires preserving its convex upper body.

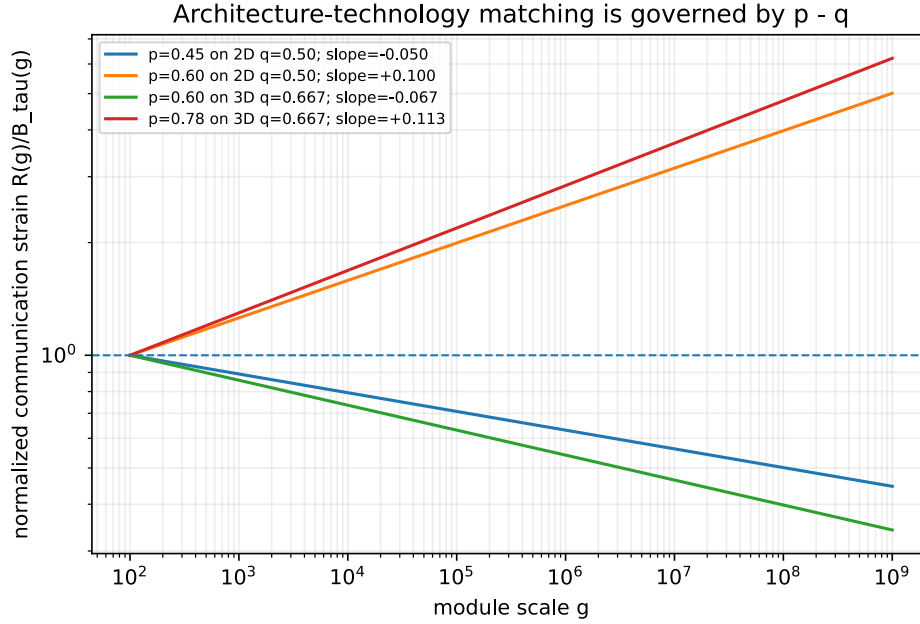


Figure 1: Normalized strain has log-log slope (p-q) in the declared model.

5 Architecture-technology scale and link matching

5.1 Scale-dependent communication strain

Let $R(g) > 0$ be the demand profile of one fixed realization and hierarchy, and let $B_\tau(g) > 0$ be the corresponding boundary-capacity profile of a technology. Their quotient

$$\chi_\tau(g) = \frac{R(g)}{B_\tau(g)}$$

measures scale-dependent strain.

Theorem 5.1 (Scaling strain). *If $R \in \text{RV}_p$ and $B_\tau \in \text{RV}_q$, then*

$$\chi_\tau \in \text{RV}_{p-q}.$$

Thus demand below the capacity index relaxes asymptotically, demand above it becomes increasingly mismatched, and equal indices are marginal, with constants and slowly varying factors deciding feasibility.

Proof. Write $R(g) = g^p \ell_R(g)$ and $B_\tau(g) = g^q \ell_B(g)$. Then $\chi_\tau(g) = g^{p-q} \ell_R(g) / \ell_B(g)$, and the quotient of positive slowly varying functions is slowly varying. $\square$

Corollary 5.2 (Homogeneous dimensional threshold). *In an ideal homogeneous d -dimensional substrate with boundary capacity $B_d(g) = \Theta(g^{(d-1)/d})$, a demand profile of index p has strain index*

$$p - \frac{d-1}{d}.$$

The marginal indices are 1/2 in two dimensions and 2/3 in three.

This is a scoped capacity statement, not an automatic routability theorem. Channel conflicts, pinning, timing, power delivery, design rules, and thermal constraints remain in Adm_τ . A fixed number of tiers changes only a constant; a tier count growing with problem size changes the model and must be registered explicitly.

5.2 Aligned links and chiplet interfaces

For one hierarchy-aligned realization, let t_e be required traffic per operation, b_e be payload per link transaction or cycle, and ϵ_e be link energy per bit.

Theorem 5.3 (Aligned-link feasibility and serialization). *In the independent fluid-link model, a one-shot embedding is feasible exactly when $t_e \leq b_e$ for every interface. The minimum uniform fluid and integral serialization factors are*

$$s^*(t, b) = \max_e \frac{t_e}{b_e}, \quad s_{\text{int}}^*(t, b) = \max_e \left\lceil \frac{t_e}{b_e} \right\rceil,$$

and the declared linear link energy is $E(t) = \sum_e \epsilon_e t_e$.

Proof. Each inequality is necessary by link capacity. It is jointly sufficient because the model declares independent links and no additional causal dependency. Uniform serialization requires $t_e \leq s b_e$ on every edge, which gives the maximum ratio and its integral ceiling. Energy adds by the declared per-bit accounting rule. $\square$

Now let one physical link unit provide payload rate B_e , b_e bits per cycle, energy ϵ_e per bit, and shoreline area a_e . Let f_{op} be the required operation rate and n_e the number of parallel link units.

Theorem 5.4 (Chiplet link-budget law). *Every embedding of the selected realization satisfies*

$$n_e \geq \left\lceil \frac{f_{\text{op}} t_e}{B_e} \right\rceil, \quad E_{e,\text{op}} \geq \epsilon_e t_e, \quad P_e \geq f_{\text{op}} \epsilon_e t_e,$$

$$A_{\text{IO}} \geq \sum_e a_e n_e, \quad L_{e,\text{ser}} \geq \left\lceil \frac{t_e}{n_e b_e} \right\rceil.$$

Any reach or shoreline limit with $n_e > n_e^{\max}$ certifies that the selected partition is infeasible under the declared package model.

Proof. The aggregate payload demand is $f_{\text{op}} t_e$, which gives the link-count bound. Per-bit accounting gives energy and power; link areas add. Parallel units carry at most $n_e b_e$ bits per cycle, yielding the serialization bound before propagation, protocol, buffering, and synchronization overhead. $\square$

The technology supplies $B_e, b_e, \epsilon_e, a_e$, reach, and protocol overheads; it does not change t_e unless the legal realization changes. UCIE rate and package classes, for example, alter the embedding coordinates without erasing a semantic bridge [35].

6 Reference intermediate representation and compiler boundary

The calculus suggests a cross-layer loop with five roles:

1. a semantic front end registers the computation and legal changes to dataflow, tiling, storage, replication, approximation, ownership, and terminal meaning;
2. a realization generator emits constructive legal classes and preserves realization identity separately from typed obligation signatures;
3. a certificate engine combines semantic lower bounds, scoped outer relaxations, and constructive witnesses;
4. a technology library distinguishes price changes from capability changes that alter the resource contract or the legal realization family; and
5. physical tools return routed or otherwise evaluated constructions as new inner witnesses and can trigger targeted challengers when a bottleneck becomes active.

A reference implementation now exercises this boundary on a real PyTorch FX symbolic trace of a Transformer microblock. The prototype emits a computation contract, a resource contract, legal transformation classes, six constructive realization classes, typed obligations, three scoped outer bounds, and nine technology/objective decision certificates. It operationalizes the left half of the CRG abstraction ladder and the dependency record needed for safe repricing and reopening. It does not automatically enumerate every legal realization, generate general converses, ingest arbitrary compiler dialects or RTL, or perform physical signoff; those remain separate research and engineering problems.

OpenROAD demonstrates autonomous RTL-to-GDS exploration, and its ArtNet flow accepts hierarchical structure and scalar Rent parameters [27, 28, 43]. A production CRG adapter would exchange typed realization records, interface-separation distributions, certified inner/outer regions, decision margins, and dependency hashes instead of one scalar exponent. The operational rule is: analyze once and reprice many times only while legality, feasibility, and certificate scope remain unchanged. A capability change reopens affected branches and requires reconstruction rather than silent reuse of an obsolete geometry.

7 Analytic component certificates

The design calculus becomes operational when a computation has both a scoped converse and constructive realizations. Dense matrix multiplication and exact attention provide complementary cases. Matrix multiplication exposes a memory–traffic–collective frontier; attention exposes a traffic–storage–recomputation frontier. Neither has a technology-independent scalar winner.

7.1 Dense matrix multiplication across memory levels

Consider classical $n \times n$ dense matrix multiplication restricted to the conventional $\Theta(n^3)$ scalar-product DAG. For a two-level memory hierarchy with M fast words, established pebble and geometric bounds give $Q(M) = \Omega(n^3/\sqrt{M} + n^2)$, while blocked algorithms attain the same order under the usual cache assumptions [30, 31].

Theorem 7.1 (Multilevel matrix-multiplication bracket). *Let $M_1 < \dots < M_L$ be the cumulative fast-side capacities exposed by an inclusive multilevel contract, and let q_i be words moved across interface i . Inputs begin beyond each modeled interface and the final output is returned beyond it. There are constants $c_0, C_0 > 0$ such that every legal classical realization satisfies*

$$q_i \geq c_0 \left(\frac{n^3}{\sqrt{M_i}} + n^2 \right) \quad (1 \leq i \leq L),$$

while one cache-oblivious recursively blocked realization simultaneously satisfies

$$q_i \leq C_0 \left(\frac{n^3}{\sqrt{M_i}} + n^2 \right) \quad (1 \leq i \leq L)$$

at every level obeying the declared ideal-cache assumptions [32]. Thus one outer orthant and one jointly realizable inner point certify the multilevel traffic projection within a constant factor.

Proof. At level i , contract every faster level into the declared cumulative M_i -word fast side and apply the two-level lower bound to that complete interface. For achievability, the same cache-oblivious recursion induces a legal blocked schedule at every ideal-cache level simultaneously. The claim therefore does not combine incompatible per-level optima or count one transfer at several levels without accounting. A noninclusive hierarchy must replace M_i by the actual capacity available on the contracted fast side. $\square$

The same structured computation also yields a scale exponent. For a set $S \subseteq [n]^3$ of $g = |S|$ multiplication triples, the discrete Loomis–Whitney inequality gives

$$|S|^2 \leq |\pi_A(S)| |\pi_B(S)| |\pi_C(S)|.$$

Hence at least one operand/output projection has size $\Omega(g^{2/3})$; cubic blocks attain $O(g^{2/3})$. The natural classical blocked realization therefore has the interior demand law

$$R(g) = \Theta(g^{2/3}).$$

This is a result for the registered structured computation, not for every Boolean circuit computing matrix multiplication.

Table 1: Declared public technology-coordinate library. Payload rates are ideal one-direction cluster projections, not protocol or contention measurements.

Technology row	Payload	Link energy	Status
UCIe-S, 32 GT/s	64 GB/s	0.50 pJ/bit	public target anchor
UCIe-A, 32 GT/s	256 GB/s	0.25 pJ/bit	public target anchor
UCIe-A, 64 GT/s	512 GB/s	0.25 pJ/bit	rate anchor; energy proxy
Local fast state	32 or 96 MiB	–	public cache-capacity anchors

Table 2: Modeled $n = 8192$, $K_{\text{chip}} = 64$ projection with two-byte words. Time uses one ideal UCIe-A32 cluster on the critical interface.

c	Memory/chiplet	Traffic/chiplet	Stages	Transfer time	Link energy
1	6.0 MiB	32.00 MiB	8	131.1 μs	2.147 mJ
2	12.0 MiB	22.63 MiB	4	92.7 μs	1.519 mJ
4	24.0 MiB	16.00 MiB	3	65.5 μs	1.074 mJ

7.2 A certified 2.5D technology map

On K_{chip} processor-chiplets with per-chiplet fast memory M , classical distributed lower bounds include a per-chiplet communication floor of order $\Omega(n^3/(K_{\text{chip}}\sqrt{M}))$ under the declared distribution assumptions [33, 31]. The 2.5D family introduces a replication factor $1 \leq c \leq K_{\text{chip}}^{1/3}$ and, up to constants and logarithmic latency terms, obeys

$$M(c) = \Theta\left(\frac{3cn^2}{K_{\text{chip}}}\right), \quad Q(c) = \Theta\left(\frac{n^2}{\sqrt{cK}}\right), \quad R(c) = O\left(\sqrt{\frac{K_{\text{chip}}}{c^3}} + \log c\right)$$

[34]. The numerical certificate declares the constructive leading model

$$q(c) = \frac{2n^2}{\sqrt{cK}}, \quad m(c) = \frac{3cn^2}{K_{\text{chip}}}, \quad r(c) = \left\lceil \sqrt{K_{\text{chip}}/c^3} \right\rceil + \lceil \log_2 c \rceil.$$

The constants are model coordinates; the exchange of memory for traffic and collective stages is inherited from the cited algorithms and converses.

For $K_{\text{chip}} = 64$, two-byte words, and $c \in \{1, 2, 4\}$, feasibility under local memory budget M_{max} is exactly

$$\frac{3cn^2}{K_{\text{chip}}}(2 \text{ bytes}) \leq M_{\text{max}}.$$

Both (qc) and (rc) decrease with c , so the largest feasible replication factor is Pareto dominant for every monotone traffic-round objective in this declared family.

The UCIe rows use public rate classes and the declared energy proxies [35, 36]; the local-state anchors follow public processor documentation [37]. Together they define a public-anchor technology library whose rows are explicit technology coordinates for the design calculus.

The resulting memory-conditioned selection is

32 MiB/chiplet	c^*
$n \leq 9459$	4
$9459 < n \leq 13377$	2
$13377 < n \leq 18918$	1
$n > 18918$	infeasible in the declared local-memory model.

With 96 MiB per chiplet, the $c = 4$ region extends to $n = 16384$. The public link rows quantify the value of a selected realization but do not change the largest-feasible- c ordering in this idealized family.

Collective capability can reverse a different pair of realizations. At $K_{\text{chip}} = 64$, a one-dimensional replicated- B point with effective collective stage count r has typed signature

$$z_{1D}(r) = (63, r, 1.03125),$$

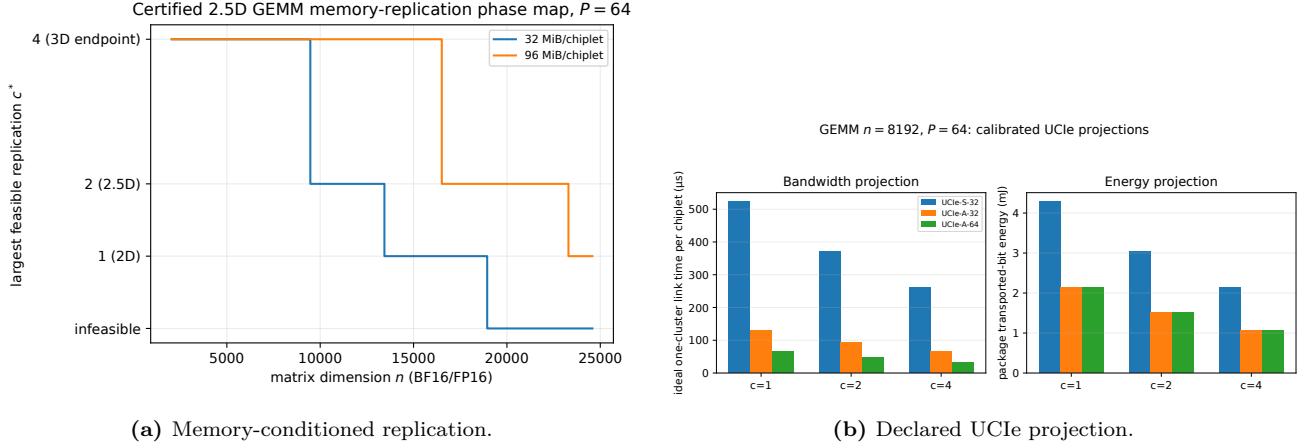


Figure 2: The 2.5D realization family is retained until memory and link technology are declared.

while the declared two-dimensional SUMMA point is $z_{2D} = (14, 8, 0.046875)$ [38, 39]. Under price $C = w_Q q + w_R r + w_M m$, the one-dimensional realization can win only for $r < 8$, and then exactly when

$$w_R > \frac{49w_Q + 0.984375w_M}{8 - r}.$$

The reversal is attached to an explicit collective capability rather than an unstated communication primitive.

7.3 Exact attention: recomputation versus storage

Let $Q, K, V \in \mathbb{R}^{N \times d}$ and consider exact attention $O = \text{softmax}(QK^T)V$ with fast memory M scalar elements. The conventional realization materializes $N \times N$ score and probability matrices; the tiled realization retains rowwise normalization state and recomputes blocks when required.

Theorem 7.2 (Imported exact-attention I/O bracket). *For $d \leq M \leq Nd$, standard exact attention has*

$$Q_{\text{std}} = \Theta(Nd + N^2), \quad S_{\text{std}} = \Theta(N^2),$$

whereas *FlashAttention achieves*

$$Q_{\text{flash}} = \Theta\left(\frac{N^2 d^2}{M}\right), \quad S_{\text{flash}} = O(N),$$

while spending additional local work through recomputation. The cited analysis proves I/O optimality over the stated SRAM range [40].

The theorem supplies incomparable typed points, not one scalar winner. For $d = 64$ and 128, the asymptotic crossover $M > d^2$ corresponds to only 8 and 32 KiB for two-byte elements, but implementation constants and additional work still require an explicit objective.

For the published A100 forward-plus-backward GPT-2-medium measurement, the reported points are

$$z_{\text{std}} = (Q, F, T) = (40.3 \text{ GB}, 66.6 \text{ GFLOP}, 41.7 \text{ ms}),$$

$$z_{\text{flash}} = (4.4 \text{ GB}, 75.2 \text{ GFLOP}, 7.3 \text{ ms})$$

[40]. Under typed additive cost $C = w_Q Q + w_F F + w_S S$, the tiled realization is preferred exactly when

$$35.9w_Q + \Delta S w_S > 8.6w_F, \quad \Delta S = S_{\text{std}} - S_{\text{flash}} > 0.$$

With zero storage price, the break-even is approximately 0.240 FLOP/byte. Public A100 peak/bandwidth ratios are approximately $838\times$ larger on the BF16 tensor-peak basis and about $52\times$ larger on the FP32 basis [41]; this establishes the arithmetic-intensity separation; measured kernel time additionally depends on utilization, scheduling, and the remaining typed coordinates.

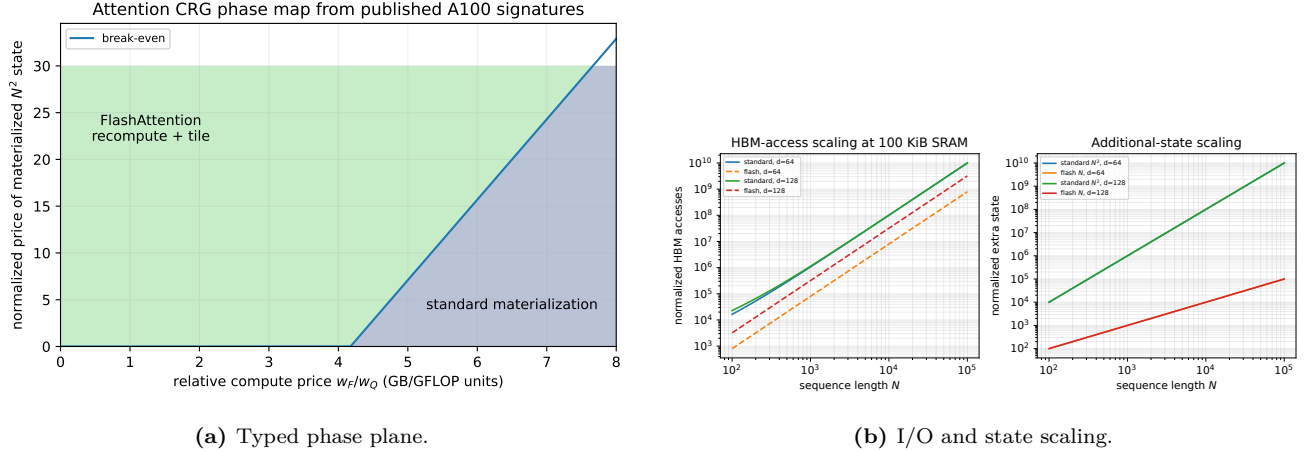


Figure 3: Traffic, work, and storage jointly determine the attention realization boundary.

7.4 Cross-stage state and layout laws

The end-to-end application joins attention and an MLP. Two general facts prevent isolated stage optima from being multiplied together without a joint resource check.

Theorem 7.3 (Static provisioning and time-shared state). *Let two isolated modes require usable state m_A and m_F on a chiplet with budget M .*

- (i) *A static fused realization with dedicated simultaneously instantiated engines is infeasible when $m_A + m_F > M$.*
- (ii) *A sequential universal realization that reuses one state pool is state-feasible when $\max(m_A, m_F) \leq M$.*

The two statements belong to different realization contracts; time sharing must expose any efficiency cost rather than treating reconfiguration as free.

Proof. Dedicated state occupies disjoint capacity and adds. State used in disjoint time intervals occupies the maximum. An embedding cannot transform one resource aggregator into the other without changing the realization. $\square$

For layout conversion, let $K_{\text{chip}} = CT = C'T'$ chiplets implement aligned two-dimensional layouts over a normalized activation tensor. Let $S_{i=1}^{K_{\text{chip}}}$ and $T_{j=1}^{K_{\text{chip}}}$ be the equal-area source and target rectangles. Each tensor element has one source and one target owner; chiplet relabeling is free and bijective. Compression, recomputation, replication, coding, cached duplicates, unequal-capacity placement, routed congestion, latency, and energy are outside this conversion contract.

Theorem 7.4 (Exact aligned-layout assignment). *After optimal chiplet relabeling, the exact retained and minimum moved fractions are*

$$r^*(C, C') = \max_{\pi \in \mathfrak{S}_{K_{\text{chip}}}} \sum_{i=1}^{K_{\text{chip}}} |S_i \cap T_{\pi(i)}|, \quad \mu^*(C, C') = 1 - r^*(C, C').$$

If $C \mid C'$ or $C' \mid C$, then

$$r^*(C, C') = \frac{\min(C, C')}{\max(C, C')}, \quad \mu^*(C, C') = 1 - \frac{\min(C, C')}{\max(C, C')}.$$

For nonnested factors the closed form need not hold.

Proof. Under a relabeling π , chiplet i retains exactly $|S_i \cap T_{\pi(i)}|$, so maximizing retention is a maximum-weight perfect assignment. Retain every selected intersection and move only each target remainder from its unique source owner; this realizes movement equal to one minus retained measure. Conversely, every element outside the retained intersections must change owner.

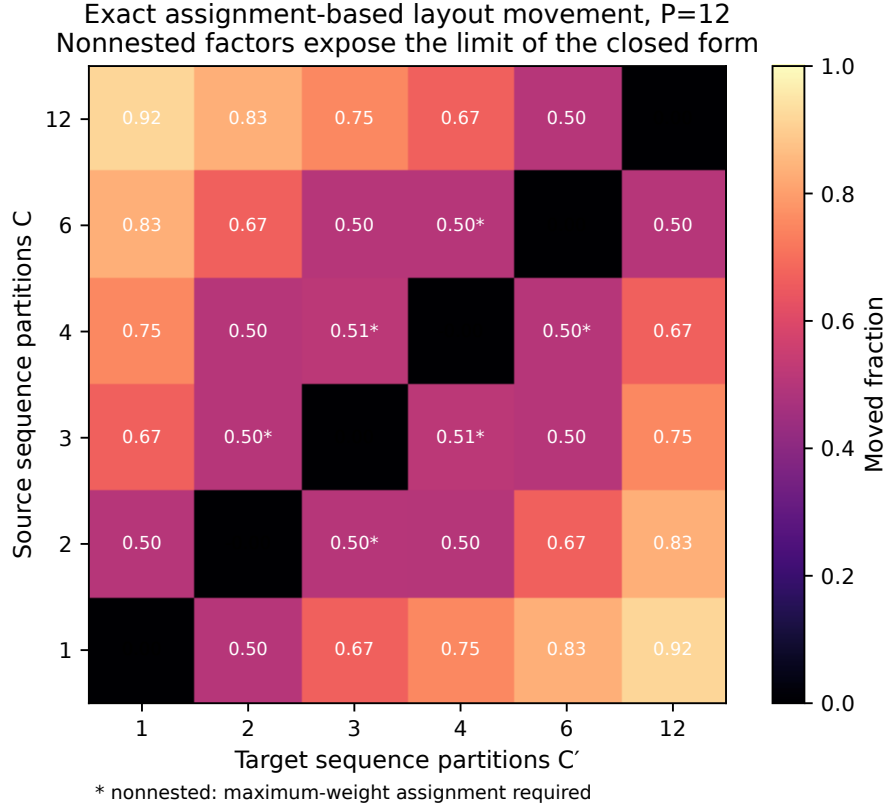


Figure 4: The $K_{\text{chip}} = 12$ nonnested example separates the general assignment law from the nested closed form.

For the nested case $C' = kC$, each nonempty intersection has area at most $1/(kK)$, so total retention is at most $1/k$. The intersection graph is k -regular and bipartite, hence has a perfect matching attaining $1/k$. The case $C > C'$ is symmetric. $\square$

At $K_{\text{chip}} = 12, C = 4, C' = 6$, the exact retained fraction is $1/2$, not $2/3$, which shows why nesting is load bearing. Auto-sharding systems already price resharding; the result here is the exact optimal-relabeling law under its declared ownership contract [17, 18].

These component certificates now supply the end-to-end case with scoped converses, constructive stage modes, state aggregators, and an exact ownership conversion law. The remaining question is which joint realization wins once all resources and package obligations are conserved simultaneously.

Part II

End-to-end certificate

8 Certified nonseparable co-design of a long-context Transformer superblock

The component certificates become decision-relevant only after dataflow, state, chiplet ownership, memory allocation, execution semantics, and package service are registered jointly. The case study considers one exact residual superblock

$$Y = X + \text{GQA}(X), \quad Z = Y + \text{SwiGLU}(Y), \quad (8.1)$$

for batch-one prefill. The conditional question is:

Given a declared workload-execution contract, a finite architecture registry, and a technology ledger, which registered class minimizes the declared steady-state initiation interval?

The answer is exhaustive for the declared finite algebraic model. The model contains the registered architecture classes, execution policies, resource aggregators, ownership obligations, integer allocations, and technology coordinates needed to derive the class decision and its phase structure.

8.1 Workload and execution contract

The primary instance uses hidden dimension $d = 4096$, feed-forward width $d_{\text{ff}} = 14336$, $h = 32$ query heads, $h_{\text{KV}} = 8$ key/value heads, two-byte BF16/FP16 elements, and contexts through $N = 131,072$ [44, 45]. A multiply-add counts as two FLOPs. If $M_{\text{exec}}(N)$ is the number of query-key pairs actually executed, the registered tensor-matmul work is

$$F_A(N) = \left(4 + 4 \frac{h_{\text{KV}}}{h}\right) N d^2 + 4 d M_{\text{exec}}(N), \quad (8.2)$$

$$F_F(N) = 6 N d d_{\text{ff}}. \quad (8.3)$$

The projection coefficient is five for the declared GQA ratio, not the MHA coefficient eight.

The three executable pair policies are

$$M_{\text{exec}}(N) = \begin{cases} N^2, & \text{dense-padded causal execution,} \\ N(N+1)/2, & \text{ideal causal-pair execution,} \\ B^2 m(m+1)/2, \quad m = \lceil N/B \rceil, & \text{block-rounded causal execution.} \end{cases} \quad (8.4)$$

The named default uses $B = 128$ and executes complete square tiles on and below the block-causal diagonal, including final-tile padding.

At $N = 16,384$, the semantic causal count is 134,225,920, the executed block-rounded count is 135,266,304, attention tensor work is 3.590593 TFLOP, and MLP tensor work is 5.772436 TFLOP. The equality $F_A = F_F$ at $N = 4d$ holds for these dimensions under dense-padded execution; it is not invariant to execution policy.

Softmax, RoPE, RMSNorm, residual additions, SwiGLU elementwise work, launch and control, synchronization, ECC, and unmodeled utilization losses are typed exclusions. They are not silently absorbed into Equations (8.2) and (8.3). Every timing claim is therefore scoped to the named tensor-work model.

One hidden-state tensor occupies

$$A(N) = N d b_{\text{elem}}, \quad b_{\text{elem}} = 2 \text{ bytes.}$$

If attention and MLP have disjoint single-owner regions, steady-state alternation moves one state to the MLP and one back to the next attention stage:

$$\boxed{Q_B(N) \geq 2 N d b_{\text{elem}}.} \quad (8.5)$$

At $N = 16,384$, $A = 128$ MiB and $Q_B = 256$ MiB. Placement can price this obligation but cannot erase it while disjoint ownership remains part of the contract.

8.2 Evidence-tiered technology contract

The declared system has $K_{\text{chip}} = 32$ compute chiplets, eight HBM placements, 8 TB/s effective aggregate HBM bandwidth, and 32 MiB usable fast state per planar fused chiplet. “Usable” is after declared software, fragmentation, and reliability derating; a witness exactly on a capacity wall has zero capacity margin.

The default dense BF16 tensor-work coordinate is 5 PFLOP/s. The corresponding 10-PFLOP/s public headline is a structured-sparse coordinate; it appears only in explicitly named sparse-peak sensitivity rows [46]. Peak is a ceiling, not delivered kernel throughput. The 8-TB/s HBM coordinate is a declared derating of public HBM capability rather than a specification for a shipping package [47].

Table 3: Declared package coordinates. Payload and energy are model coordinates anchored to public capability classes, not guarantees.

Technology	Payload	Link energy	Evidence status
Standard declared cluster	64 GB/s	0.50 pJ/bit	public-rate anchor
Advanced 32-GT/s cluster	256 GB/s	0.25 pJ/bit	declared sensitivity
Advanced 64-GT/s cluster	512 GB/s	0.25 pJ/bit	rate anchor; energy proxy
Vertical two-tier	4 TB/s	0.05 pJ/bit	declared capability study

UCIe supplies the public rate and 3D capability classes used as anchors [35]; the model does not infer aggregate payload, latency, or energy directly from raw lane rate. In the default *shared aggregate* contract, internal attention traffic, internal MLP traffic, and Equation (8.5) share one package-payload budget. A named sensitivity charges internal collectives to separate local fabrics and only the bridge to the package. The distinction is encoded in the contract name

llama31-gqa-block-rounded-causal128-dense-bf16-shared-package-aggregate-v1.

8.3 Finite architecture and mode registry

Each registered realization retains the typed signature

$$\sigma(P) = (F_{\text{loc}}, M_{\text{fast}}, Q_{\text{HBM}}, Q_{\text{pkg}}, R_{\text{coll}}, D_{\text{causal}}, E_{\text{link}}, T_{\text{II}}, T_{\text{lat}}, \Pi, \eta).$$

Capacity, traffic, bandwidth, latency, and reconfigurability remain distinct coordinates. The six architecture classes are:

H: static spatial fused.

Dedicated attention and MLP engines coexist; the stage bridge is local, but nonshareable state adds.

U: universal time-shared fused.

One reconfigurable fabric executes the stages sequentially, reuses one state pool, and pays universal attention efficiency η_U .

S: heterogeneous split.

Attention and MLP own disjoint compute and HBM pools and overlap in steady state; their internal traffic and semantic bridge consume the declared package resources.

X: dual-mode hybrid.

Split hardware also supports a universal fallback and selects the better legal U/S parent with a declared two-percent whole-path capability tax. At zero tax it ties, but cannot beat, the parent envelope without a new schedule.

M: memory-centric attention.

Attention uses a declared near-HBM, lower-compute-density capability; the MLP remains compute dense.

V: vertically specialized.

Two active tiers use twice the planar compute density, reduced internal traffic, and the declared vertical fabric.

Attention has an 8-MiB Flash-local mode and a 28-MiB collective-aware mode. The latter imports FlatAttention’s 0.893 utilization, $4.1\times$ relative speed, and $16\times$ HBM-traffic comparison as a portability hypothesis from a different MHA many-PE architecture [48]. The MLP has an 8-MiB low-replication mode and a 24-MiB $c = 4$ communication-avoiding mode. These are registered, target-specific mode coordinates within the declared realization contract.

The base prefill traffic ledger is

$$\begin{aligned}
W_A &= \left(2 + 2\frac{h_{KV}}{h}\right) d^2 b_{\text{elem}}, & W_F &= 3dd_{\text{ff}}b_{\text{elem}}, \\
Q_A^{(0)} &= W_A + 4A(N), & Q_F^{(0)} &= W_F + 6A(N), \\
D_A^{(0)} &= 2A(N), & D_F^{(0)} &= 2A(N).
\end{aligned} \tag{8.6}$$

The effective HBM and local-fabric bandwidth scales are 8 and 4 TB/s.

Table 4: Base stage-mode coordinates. Multipliers apply to Equation (8.6).

Mode	State	η	$Q/Q^{(0)}$	$D/D^{(0)}$	Peak scale
Attention Flash-local	8 MiB	0.893/4.1	16	1	1
Attention Flat-collective	28 MiB	0.893	1	0.5	1
MLP 2D-low-replication	8 MiB	0.85	1	1	1
MLP 2.5D-c4	24 MiB	0.85	0.5	0.5	1

H enumerates mode pairs whose state sum is at most 32 MiB. U enumerates pairs whose state maximum is at most 32 MiB and substitutes $\eta_U = 0.80$ for universal collective-aware attention. Thus Theorem 7.3 gives

$$28 + 24 = 52 > 32, \quad \max(28, 24) = 28 \leq 32.$$

The high-high pair is illegal in H but legal in U; U’s competitiveness is governed by its efficiency, not by a false sum of sequential state.

S fixes the collective-aware attention and $c = 4$ MLP modes and enumerates all positive integer partitions

$$(k_A, k_F) = (1, 31), \dots, (31, 1), \quad (h_A, h_F) = (1, 7), \dots, (7, 1).$$

There are $31 \times 7 = 217$ witnesses. X applies

$$T_X = \frac{\min(T_U, T_S)}{1 - 0.02}.$$

M fixes compute/HBM partitions $8/24$ and $5/3$ with a declared near-HBM attention mode. V uses the full 217-witness enumeration with twofold peak scale, half HBM traffic, one-tenth internal traffic, and the vertical package row.

When two stage modes use different aligned layouts, the exact represented-byte movement is the assignment value in Theorem 7.4; its nested closed form is used only when the partition factors are actually nested. The split class additionally carries the two-way bridge in Equation (8.5). No placement or relabeling is credited with eliminating bytes that the ownership contract requires to move.

8.4 Conserved timing and energy

For stage $j \in \{A, F\}$, let F_j, Q_j, D_j denote tensor work, HBM bytes, and internal-fabric bytes. With compute share α_j , HBM share β_j , utilization η_j , peak C_{pk} , HBM bandwidth B_{HBM} , and local fabric B_f , define

$$t_j^{\text{cmp}} = \frac{F_j}{\alpha_j C_{\text{pk}} \eta_j}, \quad t_j^{\text{HBM}} = \frac{Q_j}{\beta_j B_{\text{HBM}}}, \quad t_j^{\text{fab}} = \frac{D_j}{B_f}.$$

U reuses full resources sequentially:

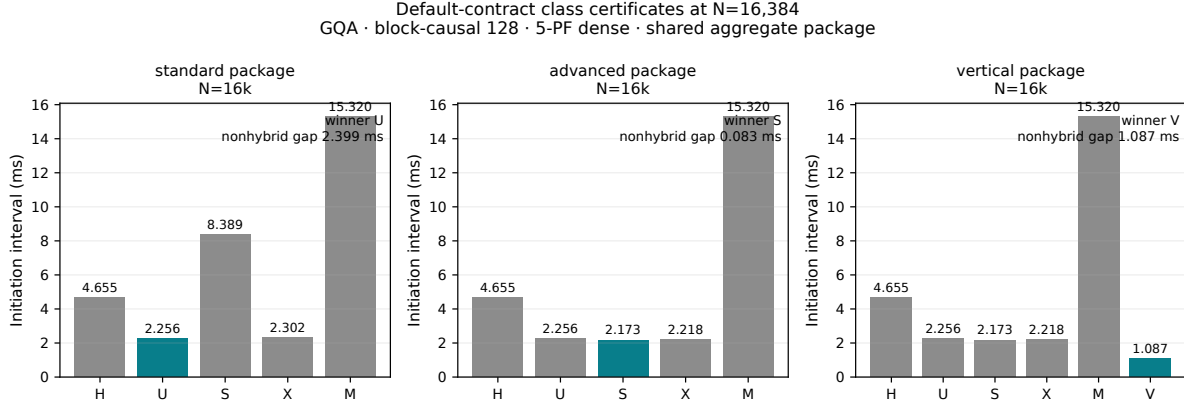
$$T_U = \max(t_A^{\text{cmp}}, t_A^{\text{HBM}}, t_A^{\text{fab}}) + \max(t_F^{\text{cmp}}, t_F^{\text{HBM}}, t_F^{\text{fab}}).$$

H is likewise registered with a sequential full-resource schedule. A hypothetical continuously overlapped stream cannot give both stages the full capacity: for full-resource demands a_r, f_r , conservation gives

$$\min_{\gamma_r \in (0,1)} \max_r \left\{ \frac{a_r}{\gamma_r}, \frac{f_r}{1 - \gamma_r} \right\} = \max_r (a_r + f_r). \tag{8.7}$$

Table 5: Default-contract class optima at $N = 16,384$, in milliseconds. X includes its two-percent tax.

Scenario	H	U	S	X	M	V	Winner / full gap
Standard 64 GB/s	4.655293	2.255868	8.388608	2.301907	15.319862	–	U / 0.046038
Advanced 512 GB/s	4.655293	2.255868	2.173152	2.217502	15.319862	–	S / 0.044350
Vertical 4 TB/s	4.655293	2.255868	2.173152	2.217502	15.319862	1.086576	V / 1.086576

**Figure 5:** Strict registered-class decisions. Thin X gaps are the declared hybrid tax; U/S parent gaps are reported separately.

Indeed each resource requires at least $a_r + f_r$, and choosing $\gamma_r = a_r / (a_r + f_r)$ attains that resource's bound. A continuous envelope is not promoted to a class witness without a common legal integer schedule.

For each S witness under shared-package accounting,

$$T_S = \max \left\{ t_A^{\text{cmp}}, t_A^{\text{HBM}}, t_F^{\text{cmp}}, t_F^{\text{HBM}}, \frac{D_A + D_F + Q_B}{B_{\text{pkg}}} \right\}. \quad (8.8)$$

Under separate-local-fabric accounting, the final term is Q_B / B_{pkg} and D_A, D_F are charged to their divided local-fabric pools. The reported S value is the exact minimum over all 217 integer witnesses. Initiation interval is not conflated with split pipeline-fill latency, which is reported separately. Package link energy is the typed coordinate

$$E_{\text{pkg}} = 8Q_{\text{pkg}}\epsilon_{\text{bit}};$$

this is the package-link energy coordinate; full-system energy is represented by additional signature coordinates.

Model precision and physical fidelity. Exact decimal values below identify the registered algebraic model and support reproducibility. The declared auxiliary-kernel exclusions can shift wall locations by amounts comparable to the printed margins. In adversarial costing, the U/S class ordering remained stable while the efficiency wall moved from about 0.881 into roughly 0.847–0.872. Prose therefore reports wall locations to about three significant figures; exact machine values remain in the released records.

8.5 Strict registered decisions and phase walls

The standard package selects U at 2.255868 ms. The advanced package selects S at 2.173152 ms, and the vertical capability selects V at 1.086576 ms. The advanced S witness has unique compute split 12/20; all 7 positive HBM partitions tie in initiation interval, with 1/7 used only as a deterministic representative. The semantic bridge takes 0.524288 ms at 512 GB/s. Under shared-package accounting, internal traffic plus the bridge totals 512 MiB per interval and 1.073742 mJ at 0.25 pJ/bit. The separately reported fill latency is 4.841878 ms.

At the advanced point, the U/S efficiency wall is

$$\eta_U^* = 0.881200. \quad (8.9)$$

The declared $\eta_U = 0.80$ is only 0.081200 below that wall. Holding other coordinates fixed, the U/S package-payload wall is 237.988577 GB/s. These unlike coordinate distances are reported separately. They are directional diagnostics, not the full-label Euclidean certificate of [Theorem 4.7](#).

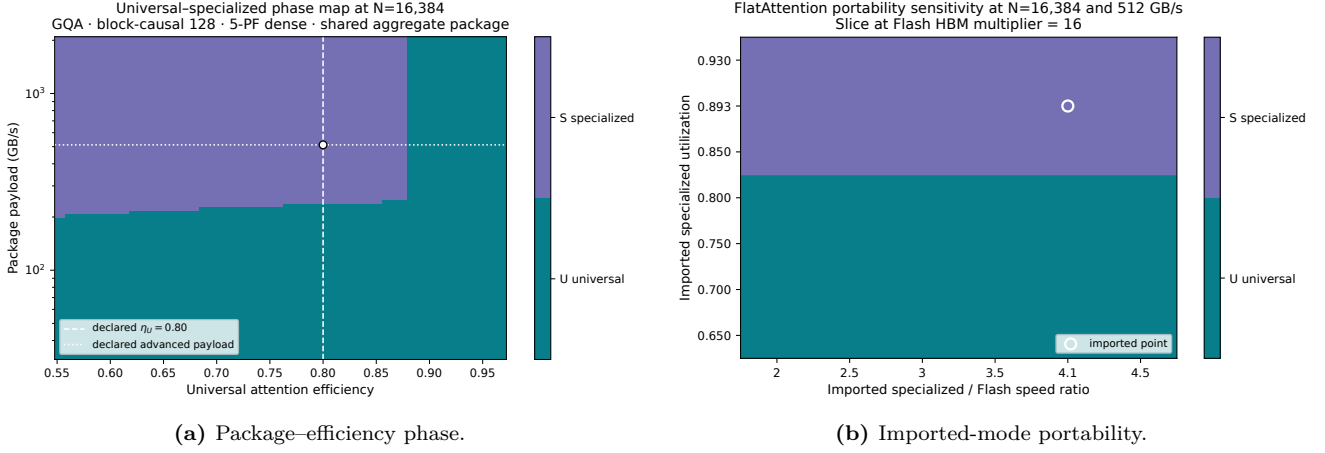


Figure 6: The advanced S point is strict but boundary-proximal.

The FlatAttention portability grid has 210 combinations: seven specialized utilizations, six speed ratios, and five Flash HBM multipliers. U wins 120 and S wins 90. The imported 0.893, $4.1\times$, $16\times$ point is a registered portability coordinate; the declared GQA realization determines its target-specific value.

The declared finite registry meets [Theorem 4.5](#): every mode and integer allocation is finite, and its timing laws are rational terms joined by finite minima and maxima. The winner, active bottleneck, and optimal witness sets are therefore piecewise constant on the semialgebraic phase cells of [Theorem 4.6](#).

8.6 Context scaling and twelve-contract sensitivity

The tensor-work profile has the form

$$R_{\text{block}}(N) = 4dM_{\text{exec}}(N) + \left[\left(4 + 4\frac{h_{KV}}{h} \right) d^2 + 6dd_{\text{ff}} \right] N.$$

For a smooth dense-padded specialization $aN^2 + \beta N$, its local exponent is

$$p_{\text{loc}}(N) = \frac{2aN^2 + \beta N}{aN^2 + \beta N},$$

which moves from one toward two. The bridge is an exponent-one mode. Its fraction decreases asymptotically, but its finite-scale affordability depends on package bandwidth, fast-state feasibility, and coefficients; exponents alone do not select the class.

Under the default contract U is strictly faster for $1 \leq N \leq 131,072$. In the exhaustive integer enumeration through 1,048,576, S is no slower on

$$[198,913, 198,926] \quad \text{and} \quad [199,041, 1,048,576].$$

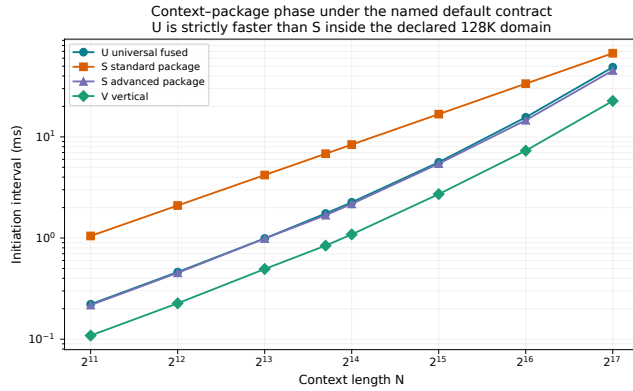
U retakes the lead on $[198,927, 199,040]$, so “sustained” refers only to the finite evaluated range and makes no claim beyond it.

The U/S/V reversal at $N = 16,384$ holds in every evaluated row, but exact walls do not: η_U^* spans 0.85 to 0.8812004, and standard-package dominance transitions move inside or beyond the declared domain. Every block-causal row has two S-no-slower intervals because tile rounding makes the predicate nonmonotone on the finite range. The machine-readable result records retain every interval rather than reporting only one crossover.

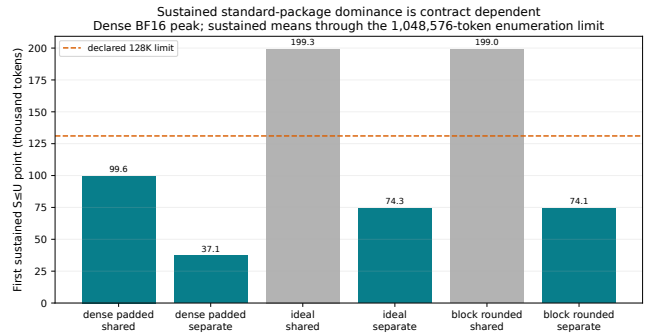
For the three primary technology scenarios, the strict argmin classes are U, S, and V. Their intersection is empty, so [Theorem 4.10](#) rules out every lossless singleton across that ledger. The family $\{U, S, V\}$ hits every argmin set and has zero class-level regret. X is not an argmin in a primary row at its declared tax; at zero tax it can hit U/S but not V. This is the exact retained-family result for the declared three-scenario technology ledger.

Table 6: Twelve-contract U/S sensitivity. The peak is either 5-PF dense or 10-PF structured-sparse sensitivity. A dagger marks an event outside the declared 128K domain.

Executed pairs / peak	Package accounting	η_U^*	Earliest	Sustained
Dense padded / 5 PF	shared aggregate	0.850000	99,641	99,641
Dense padded / 5 PF	separate local	0.850000	37,141	37,141
Dense padded / 10 PF	shared aggregate	0.850000	224,641†	224,641†
Dense padded / 10 PF	separate local	0.850000	99,641	99,641
Ideal causal / 5 PF	shared aggregate	0.877017	199,281†	199,281†
Ideal causal / 5 PF	separate local	0.877017	74,281	74,281
Ideal causal / 10 PF	shared aggregate	0.877017	449,281†	449,281†
Ideal causal / 10 PF	separate local	0.877017	199,281†	199,281†
Block causal-128 / 5 PF	shared aggregate (default)	0.881200	198,913†	199,041†
Block causal-128 / 5 PF	separate local	0.881200	73,985	74,113
Block causal-128 / 10 PF	shared aggregate	0.881200	449,025†	449,153†
Block causal-128 / 10 PF	separate local	0.881200	198,913†	199,041†



(a) Default context/package phase.



(b) Contract-sensitive transition.

Figure 7: The default has no standard-package S overtake inside 128K; gray events are finite outside-domain diagnostics.

8.7 Evidence ledger and falsification conditions

Table 7: Claim–evidence separation for the end-to-end certificate.

Class	Contents
Exact identities	GQA tensor work; pair policies; semantic bridge; layout assignment; state sum/max; conserved resources; complete integer enumeration
Imported evidence	Public model dimensions and context; FlashAttention I/O analysis; FlatAttention MHA ratios; UCle, HBM, and accelerator specifications
Declared coordinates	Usable fast state; delivered utilization; aggregate package payload and energy; universal efficiency; vertical capability; hybrid tax
Constructive witnesses	Registered schedules, allocations, initiation intervals, fill latency, package traffic, and link energy
Derived certificates	Winners, margins, phase walls, portability sweep, context intervals, and finite-ledger retention result
Unresolved calibration	Executed kernel trace; achieved GQA efficiency; fabric topology; auxiliary work; routed area/timing/power; thermal, yield, and cost

All technology coordinates are public, synthetic, or explicitly declared; each is identified by its evidence class and source contract. The primary certificate is withdrawn, weakened, or reclassified if:

- a kernel trace selects a different execution policy and the matching sensitivity changes the winner;
- the declared GQA target reaches the structured-sparse peak, or fails to approach the dense-mode efficiencies, without updating the contract;
- internal collective traffic and the bridge do not share the declared aggregate package resource;
- target measurements invalidate imported attention state, utilization, speed, or traffic coordinates;

- credible derating crosses the U/S wall or a fast-state feasibility wall;
- an omitted legal class or genuinely new hybrid schedule beats the registered envelope; or
- the objective changes from steady-state initiation interval to latency, energy, area, cost, thermal behavior, or another quantity that selects a different class.

The executable model reproduces the workload equations, mode and technology ledgers, all 217 S and V allocations, twelve-contract sweep, complete context intervals, assignment oracle, portability grid, tables, and figures. Unit tests check arithmetic identities, conservation, phase decisions, and artifact integrity. These checks establish reproducibility of the complete finite model and bind each reported decision to its equations, registry, witness enumeration, and artifact record.

The case closes the manuscript’s design chain. Stagewise-optimal modes can be jointly infeasible; time sharing restores state feasibility at an explicit efficiency price; splitting restores specialization while activating a semantic bridge; and technology changes which legal realization wins. A downstream optimizer can improve the embedding of a committed graph, but it cannot recover a realization that upstream commitment removed.

Part III

Executable realization and end-to-end validation

9 From a compiler graph to physical consequence

The formal objects of CRG admit a direct executable realization. A traced computation graph provides operator semantics, tensor dependencies, shapes, and terminal ownership. A resource contract declares legal transformations. Transformation instances generate constructive realizations. A signature extractor maps each realization to typed joint obligations. Converse modules contribute scoped outer obligations. Technology models then select and physically evaluate realizations. The reference implementation carries this chain from an exact Transformer microblock to two routed heterogeneous-package models:

$$G_{\text{FX}} \longrightarrow (F, \rho, \mathcal{T}) \longrightarrow \mathcal{P}_{\text{ref}} \xrightarrow{\sigma} \mathcal{R}_{\text{ref}} \longrightarrow \text{safe reduction} \longrightarrow \text{technology embedding.} \quad (9.1)$$

9.1 Reference CRG intermediate representation

Let G_{FX} denote the PyTorch FX symbolic trace of a microblock containing query, key, and value projections, score formation, softmax, attention-value multiplication, output projection, residual connections, feed-forward expansion and contraction, nonlinear activation, and final residual integration. The trace is converted into

$$\text{IR}_{\text{CRG}} = (F, \rho, \mathcal{T}, \mathfrak{T}, \mathcal{P}_{\text{ref}}, \sigma, \mathcal{B}_{\text{out}}), \quad (9.2)$$

where F is the terminalized computation, ρ the resource contract, $\mathcal{T}$ the ownership hierarchy, $\mathfrak{T}$ the transformation grammar, $\mathcal{P}_{\text{ref}}$ the constructive realization family, σ the typed signature extractor, and $\mathcal{B}_{\text{out}}$ the set of class-scoped outer obligations.

The computation contract records operators, tensor shapes, input/output ownership, element type, exactness, and the correspondence between graph values and terminal data. The resource contract records materialization, recomputation, fusion, split ownership, replication, time sharing, and vertical placement as legal transformation classes. Applying that grammar generates six realizations:

1. fused materialization;
2. fused recomputation;
3. split materialization;
4. split recomputation;
5. replicated split recomputation; and
6. vertical split recomputation.

For $P \in \mathcal{P}_{\text{ref}}$, the extractor emits

$$\sigma(P) = (W, M_{\text{fast}}, Q_{\text{HBM}}, Q_{\text{pkg}}, Q_{\text{bridge}}, D, L, E, A, T_{\text{max}}, Y_{\text{proxy}}, R_{\text{proxy}})(P), \quad (9.3)$$

with a unit, aggregation law, and evidence source attached to every coordinate. Three outer-obligation records constrain semantic bridge movement, state capacity, and service in their registered classes. Combining the

Reference CRG intermediate representation

A working bridge from compiler graph to proof-carrying decision

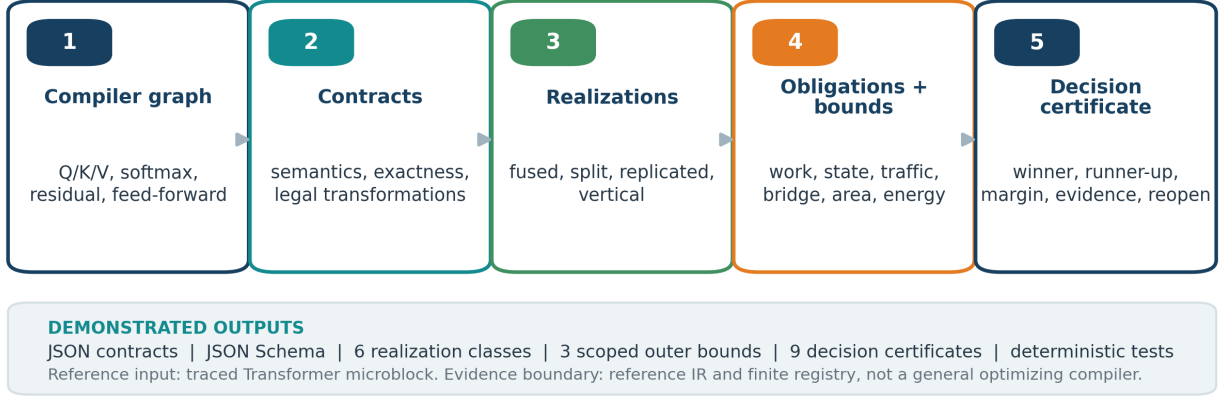


Figure 8: Reference graph-to-certificate pipeline. A traced compiler graph is expanded into computation and resource contracts, six constructive realizations, typed obligations, scoped outer bounds, and technology-conditioned decision certificates.

realization family, signatures, obligations, three technology ledgers, and three objective ledgers produces nine decision certificates. Each certificate records the selected witness, runner-up, margin, active bottleneck, coordinate provenance, and dependency set.

The three technology rows expose three distinct legal realizations: fused materialization under constrained planar service, split materialization under advanced 2.5D service, and vertical split recomputation after the vertical primitive enters the resource contract. The reference IR therefore constructs the left side of the CRG ladder rather than receiving an anonymous candidate list.

9.2 Head-to-head retention experiment

Experimental construction and interpretation. The experiment uses a declared six-realization family and a registered ledger of post-commit technology and objective shifts. The CRG declared-ledger set is constructed to intersect the argmin sets of that ledger. The comparison therefore demonstrates decision-class-aware retention against winner-only and an initial two-point Pareto projection of equal cardinality; it is not a comparison with every robust or production design-space-exploration policy.

The retention experiment fixes the same six-realization family and compares four reduction maps. Winner-only retention keeps the initial optimum. Initial-PPA Pareto retention keeps the two points nondominated under the registered initial performance, power, and area projection. CRG declared-ledger retention keeps the smallest two-realization family that intersects every argmin set in the declared future price/objective ledger. CRG universal-monotone retention keeps the five realizations whose signatures generate the full finite upper obligation region.

For a retained family $\mathcal{A}_j \subseteq \mathcal{P}_{\text{ref}}$ and post-commit technology/objective point θ , define

$$\text{Regret}_{\theta}(\mathcal{A}_j) = \min_{P \in \mathcal{A}_j} C_{\theta}(P) - \min_{P \in \mathcal{P}_{\text{ref}}} C_{\theta}(P). \quad (9.4)$$

A shifted winner is missed exactly when this quantity is positive.

A named advanced-2.5D shift changes the full-family winner from fused materialization to split materialization. Winner-only and initial-Pareto retention omit the later winner and incur 12.678% relative regret. Both CRG policies retain it and incur zero regret. Across 250 randomized post-commit changes of prices and objective weights, with legality held fixed, the result is:

Retention breadth determines post-change decision coverage

The experiment uses the same six-class registry for all policies across 250 randomized post-commit shifts.

Policy	Retained	Winner-miss rate	Mean regret
Winner-only DSE	1	58.0%	7.018%
Pareto DSE	2	57.6%	6.965%
CRG declared-ledger	2	0.4%	0.021%
CRG universal monotone	5	0.0%	0.000%

Key result: the two-realization CRG declared ledger cuts shifted-winner misses from about 58% to 0.4% while retaining no more designs than the initial Pareto baseline.

Figure 9: Head-to-head retention test. The declared CRG ledger preserves the post-commit decision at the same two-design cardinality as the initial-PPA Pareto reference policy.

Table 8: Retention under 250 post-commit changes. Regret is measured against the full six-realization family.

Policy	Retained	Shifted-winner miss rate	Mean relative regret
Winner-only reference DSE	1	58.0%	7.018%
Initial-PPA Pareto reference DSE	2	57.6%	6.965%
CRG declared ledger	2	0.4%	0.021%
CRG universal monotone	5	0.0%	0.000%

The two two-realization policies expose the content of the theorem. Their cardinality is identical; their decision information is not. The initial-PPA set preserves one starting projection, whereas the declared CRG ledger preserves the argmin structure of the stated future decision family.

A second experiment changes capability rather than price. The vertical class is absent from the original legal family and becomes admissible only after the vertical primitive is introduced. Repricing the old family yields 31.484% relative regret. Reopening the resource contract, generating the vertical realization, and rebuilding the geometry restores the global optimum. The experiment therefore distinguishes movement within a CRG slice from movement to a new slice.

9.3 Carrying one semantic obligation into routed 2.5D and 3D models

The physical study begins from one semantic statement: disjoint ownership of the two stages requires the hidden state to cross their boundary. Two technology contracts serve the same obligation through different embeddings: a side-by-side 2.5D specialist realization and a face-to-face 3D specialist realization. A negotiated A* router uses a 0.5 mm package grid and four routing layers.

For routed net bundle e , the compact timing model is

$$L_e = 10^9 \frac{q_e}{B_e} + L_e^{\text{prop}} + L_e^{\text{proto}}, \quad L_e^{\text{prop}} = \begin{cases} 0.015 \ell_e, & e \text{ planar,} \\ 0.02, & e \text{ vertical,} \end{cases} \quad (9.5)$$

where q_e is bytes per operation, B_e is allocated bytes per second, and ℓ_e is routed planar length in millimeters; the factor 10^9 expresses serialization time in nanoseconds. Link power is the transported bit rate times the declared energy per bit. Signal bumps and port-row count determine required shoreline.

The steady-state thermal field solves the finite-difference conductance system

$$G_{\text{th}} T = P + G_{\text{amb}} T_{\text{amb}}, \quad (9.6)$$

with registered lateral, vertical, and ambient conductances. The PDN field solves the resistive mesh system

$$G_{\text{pdn}} v = s - i, \quad (9.7)$$

with edge supply and a declared vertical-bump delivery drop for upper tiers. For a component of area A , defect density D_0 , and clustering parameter α , the negative-binomial yield coordinate is

$$Y(A; D_0, \alpha) = \left(1 + \frac{D_0 A}{\alpha}\right)^{-\alpha}, \quad (9.8)$$

and system yield multiplies die, interposer, attachment, and vertical-bond terms. The test coordinate is normalized access-lane service with a stack-depth factor. The reliability coordinate combines Arrhenius temperature acceleration

$$A_F(T) = \exp\left[\frac{E_a}{k_B} \left(\frac{1}{T_{\text{ref}}} - \frac{1}{T}\right)\right] \quad (9.9)$$

with routed-interconnect and stack penalties.

Table 9: Deterministic routed comparison for the same semantic ownership contract.

Metric	2.5D side-by-side	3D face-to-face
Package area	1444 mm ²	1122 mm ²
Active silicon	528 mm ²	512 mm ²
Modeled power	278.63 W	274.53 W
Semantic-bridge latency	2002.11 ns	256.62 ns
Critical-link latency	2002.58 ns	2002.13 ns
Maximum temperature	49.91°C	54.04°C
Required shoreline	3.658 mm	1.467 mm
Test-observability score	1.000	0.905
Yield proxy	0.455	0.494

The vertical service reduces bridge latency by 87.2%, shoreline by 59.9%, and package area by 22.3%. HBM service remains near 2002 ns in both realizations, so the critical-link latency remains essentially unchanged. At the same time, vertical integration raises the maximum modeled temperature, concentrates routes, reduces test observability, and changes the reliability-risk coordinate. The physical result is therefore geometric: improving the service of one semantic obligation exposes a different active face of the joint resource system.

9.4 Integrated computation-to-technology statement

The three reference results instantiate the theoretical architecture:

$$\underbrace{\text{IR construction}}_{(F, \rho, T), \mathcal{P}, \sigma, \mathcal{R}} \longrightarrow \underbrace{\text{safe reduction}}_{\Gamma_{\mathcal{A}}, \mathcal{K}_{\mathcal{A}}} \longrightarrow \underbrace{\text{physical selection}}_{\tau, \phi, \Phi_{\tau}}. \quad (9.10)$$

The graph-to-certificate implementation constructs the legal family and typed signatures; the retention experiment tests decision sufficiency under post-commit change; and the routed study maps one semantic obligation into coupled physical coordinates. They form a complete executable reference path through computation-realization geometry.

10 Conclusion

Computation-realization geometry turns architecture selection into a proof-carrying decision. The selected realization is accompanied by the computation and resource contracts, competing legal classes, typed obligations, outer bounds, constructive witnesses, margin, active bottleneck, and dependency set. This record identifies both the reason for commitment and the exact conditions under which the decision is repriced or reopened.

The design calculus closes the path from joint resource geometry to physical selection. Pareto order, semialgebraic phase cells, certified inner/outer brackets, argmin-hitting retention, scaling strain, and link budgets operate on the same realization bundle. Component laws and the Transformer registry show how local stage choices become a nonseparable system decision.

The reference implementation establishes the corresponding infrastructure path. A traced compiler graph becomes a computation contract, legal transformation classes, six realization witnesses, typed obligations, scoped

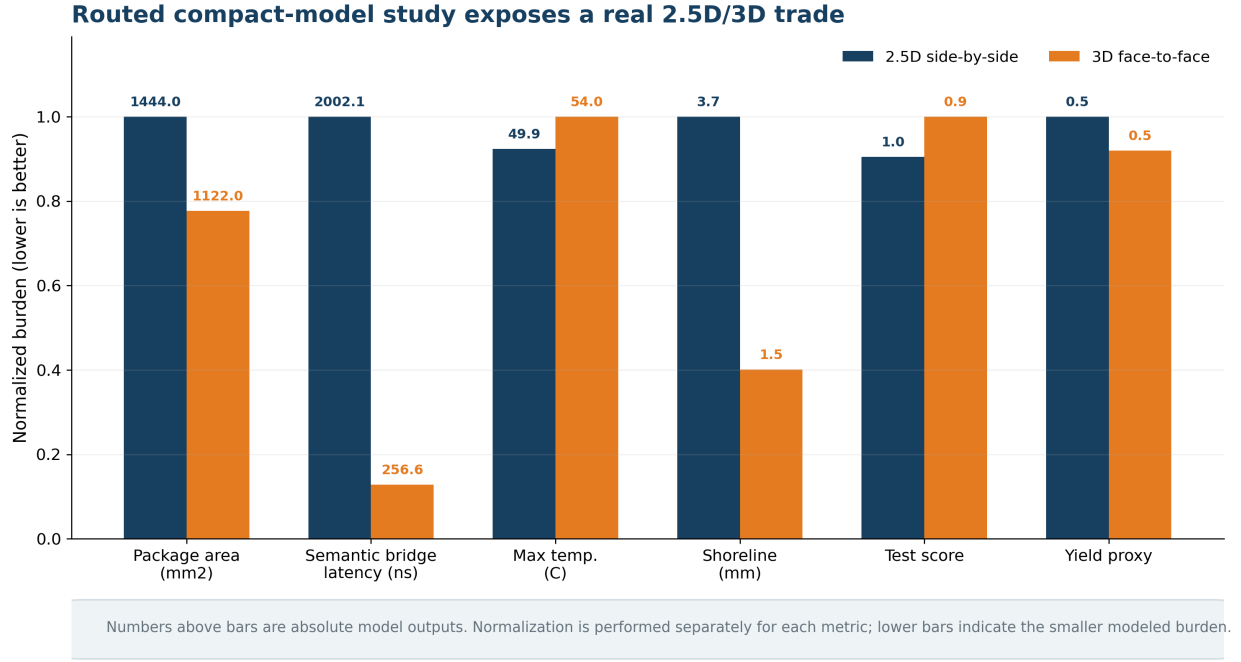


Figure 10: Routed heterogeneous-integration tradeoff. The vertical embedding changes bridge service, shoreline, area, temperature, test, and reliability coordinates while HBM remains the active critical path.

outer bounds, and nine decision certificates. The retention experiment tests the Safe-Commitment Law directly. The routed 2.5D/3D study shows that a large improvement to one semantic bridge can leave HBM as the active end-to-end bottleneck while changing area, shoreline, thermal behavior, test access, and reliability.

CRG therefore provides a common scientific interface among compilers, architecture exploration, package technology, and physical design: construct the legal geometry once, preserve the quotient required by future decisions, and return every new physical result as a witness in the same versioned object.

References

- [1] K. Keutzer, S. Malik, A. R. Newton, J. M. Rabaey, and A. Sangiovanni-Vincentelli, “System-level design: Orthogonalization of concerns and platform-based design,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, no. 12, pp. 1523–1543, 2000, doi:10.1109/43.898830.
- [2] A. Sangiovanni-Vincentelli, L. P. Carloni, F. De Bernardinis, and M. Sgroi, “Benefits and challenges for platform-based design,” in *Proceedings of the 41st Design Automation Conference*, pp. 409–414, 2004.
- [3] L. de Alfaro and T. A. Henzinger, “Interface automata,” in *Proceedings of the 8th European Software Engineering Conference*, pp. 109–120, 2001, doi:10.1145/503209.503226.
- [4] A. Benveniste, B. Caillaud, D. Nickovic, R. Passerone, J.-B. Raclet, P. Reinkemeier, A. Sangiovanni-Vincentelli, W. Damm, T. A. Henzinger, and K. G. Larsen, *Contracts for System Design*. Foundations and Trends in Electronic Design Automation, vol. 12, no. 2–3, pp. 124–400, 2018, doi:10.1561/10000000053.
- [5] C. Mead and L. Conway, “VLSI systems design: A new dimension of computer education,” *Proceedings of the IEEE*, vol. 67, no. 9, pp. 1225–1234, 1979.
- [6] C. Mead and L. Conway, *Introduction to VLSI Systems*. Addison-Wesley, 1980.
- [7] U. Bondhugula, A. Hartono, J. Ramanujam, and P. Sadayappan, “A practical automatic polyhedral parallelizer and locality optimizer,” in *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation*, 2008, pp. 101–113.

- [8] J. Ragan-Kelley et al., “Halide: A language and compiler for optimizing parallelism, locality, and recomputation in image processing pipelines,” in *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation*, 2013, pp. 519–530.
- [9] T. Chen et al., “TVM: An automated end-to-end optimizing compiler for deep learning,” in *13th USENIX Symposium on Operating Systems Design and Implementation*, 2018, pp. 578–594.
- [10] C. Lattner et al., “MLIR: Scaling compiler infrastructure for domain specific computation,” in *IEEE/ACM International Symposium on Code Generation and Optimization*, 2021, pp. 2–14.
- [11] R. Tate, M. Stepp, Z. Tatlock, and S. Lerner, “Equality saturation: A new approach to optimization,” in *Proceedings of the 36th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, 2009, pp. 264–276.
- [12] M. Willsey et al., “egg: Fast and extensible equality saturation,” *Proceedings of the ACM on Programming Languages*, vol. 5, no. POPL, Art. 23, 2021.
- [13] A. Parashar et al., “Timeloop: A systematic approach to DNN accelerator evaluation,” in *IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, 2019. https://research.nvidia.com/publication/2019-03_timeloop-systematic-approach-dnn-accelerator-evaluation.
- [14] H. Kwon, P. Chatarasi, M. Pellauer, A. Parashar, V. Sarkar, and T. Krishna, “Understanding reuse, performance, and hardware cost of DNN dataflows: A data-centric approach using MAESTRO,” in *IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 754–768, 2019. <https://arxiv.org/abs/1805.02566>.
- [15] L. Mei, P. Houshmand, V. Jain, S. Giraldo, and M. Verhelst, “ZigZag: Enlarging joint architecture–mapping design space exploration for DNN accelerators,” *IEEE Transactions on Computers*, vol. 70, no. 8, pp. 1160–1174, 2021. <https://arxiv.org/abs/2007.11360>.
- [16] Q. Huang et al., “CoSA: Scheduling by constrained optimization for spatial accelerators,” in *ACM/IEEE International Symposium on Computer Architecture (ISCA)*, 2021. <https://arxiv.org/abs/2105.01898>.
- [17] L. Zheng et al., “Alpa: Automating inter- and intra-operator parallelism for distributed deep learning,” in *OSDI*, pp. 559–578, 2022.
- [18] D. Lepikhin et al., “GSPMD: General and scalable parallelization for ML computation graphs,” arXiv:2105.04663, 2021. <https://arxiv.org/abs/2105.04663>.
- [19] J. Cai et al., “Gemini: Mapping and architecture co-exploration for large-scale DNN chiplet accelerators,” in *IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, 2024. <https://arxiv.org/abs/2312.16436>.
- [20] H. Qu et al., “MLDSE: Scaling design space exploration infrastructure for multi-level hardware,” arXiv:2503.21297, 2025.
- [21] H. Jin et al., “Mozart: A chiplet ecosystem–accelerator codesign framework for composable bespoke application specific integrated circuits,” arXiv:2510.08873, 2025.
- [22] R. T. Rockafellar, *Convex Analysis*. Princeton University Press, 1970.
- [23] H. P. Benson, “An outer approximation algorithm for generating all efficient extreme points in the outcome set of a multiple objective linear programming problem,” *Journal of Global Optimization*, vol. 13, pp. 1–24, 1998.
- [24] C. H. Papadimitriou and M. Yannakakis, “On the approximability of trade-offs and optimal access of Web sources,” in *Proceedings of the 41st IEEE Symposium on Foundations of Computer Science*, 2000, pp. 86–92.
- [25] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust Optimization*. Princeton University Press, 2009.
- [26] Y. N. Wu, J. S. Emer, and V. Sze, “Accelergy: An architecture-level energy estimation methodology for accelerator designs,” in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2019. <https://accelergy.mit.edu/>.

- [27] A. B. Kahng, “OpenROAD: Toward a self-driving, open-source digital layout implementation tool chain,” in *Proceedings of the ACM International Symposium on Physical Design*, 2019.
- [28] The OpenROAD Project, “OpenROAD documentation,” accessed July 2026. <https://openroad.readthedocs.io/>.
- [29] Y. Shi et al., “Open3DBench: Open-source benchmark for 3D-IC backend implementation and PPA evaluation,” arXiv:2503.12946, 2025, revised 2026.
- [30] J.-W. Hong and H. T. Kung, “I/O complexity: The red-blue pebble game,” in *Proceedings of the 13th ACM Symposium on Theory of Computing*, pp. 326–333, 1981.
- [31] G. Ballard, J. Demmel, O. Holtz, and O. Schwartz, “Minimizing communication in numerical linear algebra,” *SIAM Journal on Matrix Analysis and Applications*, vol. 32, no. 3, pp. 866–901, 2011.
- [32] M. Frigo, C. E. Leiserson, H. Prokop, and S. Ramachandran, “Cache-oblivious algorithms,” in *Proceedings of the 40th IEEE Symposium on Foundations of Computer Science*, pp. 285–297, 1999.
- [33] D. Irony, S. Toledo, and A. Tiskin, “Communication lower bounds for distributed-memory matrix multiplication,” *Journal of Parallel and Distributed Computing*, vol. 64, no. 9, pp. 1017–1026, 2004.
- [34] E. Solomonik and J. Demmel, “Communication-optimal parallel 2.5D matrix multiplication and LU factorization algorithms,” in *Euro-Par 2011*, pp. 90–109, 2011.
- [35] UCIE Consortium, “UCIE 3.0 Specification,” 2025. <https://www.uciexpress.org/specifications>.
- [36] UCIE Consortium, “UCIE 2.0 White Paper,” 2024, accessed through the official specification portal. <https://www.uciexpress.org/specifications>.
- [37] AMD, “4th Gen AMD EPYC Processor Architecture,” white paper, 2023. <https://www.amd.com/content/dam/amd/en/documents/products/epyc/4th-gen-epyc-processor-architecture-white-paper.pdf>.
- [38] R. A. van de Geijn and J. Watts, “SUMMA: Scalable universal matrix multiplication algorithm,” *Concurrency: Practice and Experience*, vol. 9, no. 4, pp. 255–274, 1997.
- [39] R. W. Hockney, “The communication challenge for MPP: Intel Paragon and Meiko CS-2,” *Parallel Computing*, vol. 20, no. 3, pp. 389–398, 1994.
- [40] T. Dao, D. Y. Fu, S. Ermon, A. Rudra, and C. Ré, “FlashAttention: Fast and memory-efficient exact attention with IO-awareness,” in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 16344–16359. Available: <https://arxiv.org/abs/2205.14135>.
- [41] NVIDIA, “NVIDIA A100 Tensor Core GPU,” product datasheet and architecture white paper, 2020–2021. <https://www.nvidia.com/en-us/data-center/a100/>.
- [42] J. Bochnak, M. Coste, and M.-F. Roy, *Real Algebraic Geometry*. Springer, 1998.
- [43] S. Kang et al., “ArtNet: Hierarchical clustering-based artificial netlist generator for ML and DTCO applications,” preprint arXiv:2510.13582, 2025.
- [44] A. Grattafiori et al., “The Llama 3 herd of models,” arXiv:2407.21783, 2024. Available: <https://arxiv.org/abs/2407.21783>.
- [45] Meta Llama, “Llama 3 reference implementation and model parameters,” 2024. <https://github.com/meta-llama/llama3>.
- [46] NVIDIA, “GB200 NVL72,” product information, accessed July 2026. <https://www.nvidia.com/en-us/data-center/gb200-nvl72/>.
- [47] Micron Technology, “Micron HBM3E,” product information, accessed July 2026. <https://www.micron.com/products/memory/hbm/hbm3e>.
- [48] C. Zhang et al., “FlatAttention: Dataflow and fabric collectives co-optimization for efficient multi-head attention on tile-based many-PE accelerators,” arXiv:2505.18824, 2025. <https://arxiv.org/abs/2505.18824>.

- [49] R. Bojanić and E. Seneta, “A unified theory of regularly varying sequences,” *Mathematische Zeitschrift*, vol. 134, pp. 91–106, 1973.
- [50] J. Galambos and E. Seneta, “Regularly varying sequences,” *Proceedings of the American Mathematical Society*, vol. 41, no. 1, pp. 110–116, 1973.
- [51] J. Hajnal, “Weak ergodicity in non-homogeneous Markov chains,” *Proceedings of the Cambridge Philosophical Society*, vol. 54, no. 2, pp. 233–246, 1958.
- [52] J. E. Cohen, “Ergodic theorems in demography,” *Bulletin of the American Mathematical Society*, vol. 1, no. 2, pp. 275–295, 1979.
- [53] E. Seneta, *Non-negative Matrices and Markov Chains*, revised ed. Springer, 2006.
- [54] D. J. Hartfiel, *Nonhomogeneous Matrix Products*. World Scientific, 2002.
- [55] S. Williams, A. Waterman, and D. Patterson, “Roofline: An insightful visual performance model for multicore architectures,” *Communications of the ACM*, vol. 52, no. 4, pp. 65–76, 2009.
- [56] J. Symons et al., “Stream: Layer-fused multi-core dataflow exploration for deep neural networks,” 2023.
- [57] Y. S. Shao et al., “Simba: Scaling deep-learning inference with multi-chip-module-based architecture,” in *Proceedings of MICRO*, 2019.
- [58] E. T. Kalns and L. M. Ni, “Processor mapping techniques toward efficient data redistribution,” *IEEE Transactions on Parallel and Distributed Systems*, 1995.
- [59] A. Schrijver, *Theory of Linear and Integer Programming*. Wiley, 1986.
- [60] R. M. Karp, “Reducibility among combinatorial problems,” in *Complexity of Computer Computations*, pp. 85–103, 1972.
- [61] J. H. Davenport and J. Heintz, “Real quantifier elimination is doubly exponential,” *Journal of Symbolic Computation*, vol. 5, pp. 29–35, 1988.
- [62] E. Kushilevitz and N. Nisan, *Communication Complexity*. Cambridge University Press, 1997.

Collective attribution, accountability, and AI research provenance

Fitih M. Cinnor initiated, directed, and led the research program; established its scientific objectives and evidentiary standards; selected and adjudicated the claims advanced here; and reviewed the supporting mathematics and computation. He is the named and accountable human author, responsible for the accuracy, integrity, scope, and correction of the work. Accountable correspondence: fcinnor@semiaifoundry.com.

semiAIfoundry Research, a research group of Semi AI Foundry LLC, is the collective scholarly attribution for the coordinated research effort. It recognizes integrated intellectual work across human, artificial, and hybrid forms of intelligence rather than masking those contributions behind an exclusively individual account. Collective attribution does not displace human accountability or create AI legal authorship, ownership, signatory authority, representation authority, or accountable-person status.

The research used a human-directed, orchestrated multi-agent process. AI systems made material contributions to candidate generation, mathematical derivation, theorem formalization, counterexample construction, numerical implementation, software testing, high-precision and interval-oriented verification, literature mapping, prior-art comparison, research-record synthesis, adversarial critique, visualization, drafting, editing, and publication engineering. OpenAI GPT-5.6 Pro receives explicit credit for cross-corpus synthesis, mathematical exploration, derivation checking, theorem formalization, counterexample search, software and verifier development, exposition, consistency review, and manuscript construction. Anthropic Claude systems, including Fable 5, receive explicit credit for mathematical exploration, numerical-instrument development, adversarial review, gap identification, and editorial critique. The Fable 5 review was an internal AI-assisted audit, not independent external peer review or third-party validation.

These credits record provenance, not scientific status. AI-generated material acquires no standing merely by being generated. Scientific standing comes from proofs, computations, verification controls, independent reproductions, source checks, and correction records. No AI system is an author, legal owner, signatory, authorized representative, or accountable person. No identification or credit implies that OpenAI, Anthropic, EPFL, an artifact source, or any other provider or institution owns, sponsors, approves, warrants, or endorses this work, or accepts responsibility for its claims. Third-party rights remain governed by the applicable notices. The research program received no external funding. The accountable human author declares no competing interests or conflicts of interest.

Copyright, licensing, and artifact availability

Copyright © 2026 Semi AI Foundry LLC. Except where otherwise noted, the manuscript and project-created figures in the exact versioned public payload are licensed under CC BY 4.0 (<https://creativecommons.org/licenses/by/4.0/>), with requested attribution to “Fitih M. Cinnor for semiAIfoundry Research.” Reusers must give appropriate attribution, provide the license link, and indicate changes; attribution does not imply endorsement. Project-created code is licensed under Apache-2.0. Third-party materials remain governed by their recorded upstream terms. The licensed payload is identified by its versioned source, results, artifacts, version number, public file index, root SHA-256 manifest, and archival deposit. These grants apply only to the files and version identified in that payload and do not extend to other releases or evolving research-program content.